Chapter 3

Clustering microarray data

The first level of interest for molecular biologists is to identify genes whose expression level is significantly changed under different experimental conditions. Basic statistical techniques can be applied to solve this problem efficiently [7, 83, 72, 70]. However, such an analysis treats the genes as individuals rather than exploring their relation with each other. On the other hand, for every gene, the detailed information about its expression profile as a whole over all the experiment under study is neglected in this first-level analysis. To make better use of the full-scale information provided by microarray experiments, the next level of insight is provided by clustering genes into biological meaningful groups according to their pattern of expression. Comparing with the full data itself, such groups of related genes are much more tractable for further studies including gene function and regulation.

Based on the assumption that expressional similarity (i.e. coexpression) implies functional similarity of the genes (and vice versa), the challenge of finding genes that might be involved in the same biological process is thus transformed to the problem of clustering genes into groups based on their similarity in expression profiles. Genes that have similar expression profiles are said to be coexpressed.

The first generation of clustering algorithms (e.g., hierarchical clustering [25], k-means [32] and self-organizing maps (SOM) [45]) applied to gene expression profiles were mostly developed outside biological research. Although encouraging results have been produced [71, 75, 73], some of their characteristics often complicate their use for clustering expression data [68]. They require, for example, the predefinition of one or more user-defined parameters that are hard to estimate by a biologist (e.g., the predefinition of number of clusters in k-means and SOM – this number is almost impossible to predict in advance). Moreover, changing these parameter settings will often have a strong impact on the final result. These methods therefore need extensive parameter fine-tuning, which means that a comparison of the results with different parameter settings is almost always necessary – with the additional difficulty that comparing the

quality of the different clustering results is hard. Another problem is that the first-generation algorithms often force every data point into a cluster. In general, a considerable number of genes included in the microarray experiment do not contribute to the biological process studied, and these genes will therefore have seemingly constant or even random expression profiles rather than having similar expression profiles with the other genes. Including these noisy genes in one of the clusters will contaminate their content and make these clusters less suitable for further analysis. Finally, the computational and memory complexity of some of these algorithms often limit the number of expression profiles that can be analyzed at once.

In addition to the above limitations of the first-generation clustering algorithms, the specific characteristics of microarray data, such as the high dimensionality, the highly noisy measurements, the complex biological process hidden behind, in general has created the need for clustering methods particularly tailored. Accordingly, desired features for microarray data cluster analysis include fast calculation speed, robustness, determinism, easy interpretation, and so on.

A second generation of clustering algorithms have started to tackle some of the limitations of the earlier methods, while seeking to meet those specific requirements for microarray data. These algorithms include model-based algorithms [93, 31, 52], the self-organizing tree algorithm [35], quality-based algorithms [36, 69], simulated annealing [50], the cluster affinity search technique [13], and biclustering algorithms [19, 67, 33, 11]. Also, some procedures were developed that could help biologists to estimate some of the parameters needed for the first generation of algorithms (such as the number of clusters present in the data [31, 50, 93]).

In relation with the thesis, we first give an general reminder of the use of classical clustering methods, namely hierarchical clustering, k-means clustering, and SOM, on microarray data. We then focus the discussion of the second-generation clustering algorithms for microarray data analysis on the model-based algorithms. Finally, we provide an overview on the current status of researches on biclustering algorithms for microarray data. To conclude this chapter, a list of popular methods to assess the quality of the clusters for microarray data is provided.

3.1 Some preliminaries

Before we dive into the detailed discussion of various clustering algorithms, there are yet a couple of common issues shared by different clustering techniques.

3.1.1 Clustering genes vs. clustering experiments

So far, we have addressed the clustering problem of microarray data only as an analysis to find genes that behave similarly over the experimental conditions. Instead of clustering genes, we can also cluster experimental conditions, where

the task is to find groups of experiment conditions (which can be, for example, tumor samples) across which the all the genes behave similarly. This types of clustering can be helpful for problems like histopathological discovery of tumors whose types are primarily unknown.

To put the clustering problems in a mathematical language, when we cluster genes, each expression profile is often represented as a data point (or a vector, pointing from the origin) in a high-dimensional space whose dimension equals to the number of experiments presented in the study. The Euclidean axis of the data point is given by its expression value under every experiment. Similarly, when experiments are to be clustered, it is the experiments that are presented as data points (or vectors) in a Euclidean space whose dimension equals the number of genes in the data set. In both of the cases, the problem then is to find clusters of the data points in these high dimensional spaces. We will based our discussion hereafter mainly on the clustering of gene expression profiles, but the same methodologies can also be applied to the clustering of experiments.

3.1.2 Standardization of gene expression profiles

Biologists are mainly interested in grouping gene expression that have the same relative behavior; i.e., genes that are up- and downregulated together. Genes showing the same relative behavior but with diverging absolute behavior (e.g., gene expression profiles with a different baseline or a different amplitude but going up and down at the same time) will have a relatively high Euclidean distance (see the next section for details). Cluster algorithms based on this distance measure will therefore wrongfully assign the genes to different clusters. This effect can largely be prevented by applying standardization or rescaling to the gene expression profiles to have zero mean and unit standard deviation. Gene expression profiles showing the same relative behavior will have a small(er) Euclidean distance after rescaling [58].

3.1.3 Distance metrics

Depending on the way to define a cluster, clustering methods can be divided into two types – model-based clustering methods and distance-based clustering methods. Model-based clustering methods consider the data points in the high-dimensional space under study as a whole. Their aim is to discover the underlying structure of these data points. To be more specific, model-based clustering algorithms assume that the data points in the high-dimensional space as generated by a mixture of probabilistic models with different parameters. Each of these models is thus defined as a cluster.

Distance-based clustering methods (to which most of the classical clustering methods, such as hierarchical clustering, k-means, and SOM belong), on the other hand, explore the pairwise relation between the data points in the high-dimensional space under study. In their case, a cluster contains data points whose pairwise distance between one another is lower than a threshold value. Here, the problem arises from how the distance should be defined. Hereun-

der, we give a brief overview of common distance metrics applied for clustering microarray data.

1. *Pearson correlation:* This is the most commonly used distance metric. It is often denoted by r. Pearson correlation is the is the cosine between two vectors. This means that it measures the similarity in the shapes of two profiles, while not taking the magnitude of the profiles into account. Therefore, Pearson correlation suits well the intuition of biologists for what it means that two expression profiles to be "coexpressed" [25].

2. *Squared Pearson correlation:* This is the squared product of Pearson correlation. Therefore, squared Pearson correlation r^2 neglects the sign in front of a Pearson correlation measure and considers two vectors pointing to the exact opposite directions to be perfectly similar (i.e., in this case, $r = -1$ while $r^2 = 1$). This character makes the squared Pearson correlation able to capture inverse relationships among gene expression profiles, which might also be interesting for biologists.

3. *Euclidean distance:* Euclidean distance measures the absolute distance between two points in space. That is, it is the length of the straight line connecting the two points. As mentioned previously, the Euclidean distance measures the similarity between the absolute behaviors of genes, while the biologists are more interested in their relative behaviors. Thus, when using Euclidean distance metric, a standardization procedure is needed before clustering can be applied to the gene expression profiles. Note here that after the standardization, the Euclidean distance is related to the Pearson correlation between two points x and y by $|x - y|^2 = 2(1 - |r|)$ [4].

4. *Jackknife correlation:* New techniques of distance measurements are developed to overcome some of the shortcomings of the classical distance metrics. One of the examples is the jackknife correlation [36], which is an improvement for Pearson correlation measure. Although Pearson correlation gives high score to coexpressed genes, it is not robust to outliers. An extreme case of a false positive is, for example, when two genes are totally unrelated under all but one experimental conditions, where they both have a high peak. These high peaks then pull the vector presentations of the two profiles to the same direction, and therefore the two genes will have a high Pearson correlation. Jackknife correlation increases the robustness to single outliers by computing a collection of all the possible leave-one-(experiment)-out Pearson correlation between two genes and then select the minimum of the collection as the final measure for the correlation.

3.2 Classical clustering methods

3.2.1 Hierarchical clustering

The first introduction of hierarchical clustering to the world of biology was its application to the construction of phylogenetic trees. Early applications of the method to gene expression data analysis [25, 71] have proved its usefulness.

Hierarchical clustering has almost become the *de facto* standard for gene expression data analysis, probably because of its intuitive presentation of the clustering results. The whole clustering process is presented as a tree called a dendrogram, the original data are often reorganized in a heat map demonstrating the relationships between genes or conditions.

Concerning the clustering procedure, two approaches to hierarchical clustering are possible: divisive clustering (a top-down approach as is used in [4]), and agglomerative clustering (a bottom-up approach, see for example [25]). In agglomerative clustering [25], each expression profile is initially assigned to one cluster; at each step, the distance between every pair of clusters is calculated and the pair of clusters with the minimum distance is merged; the procedure is carried on iteratively until a single cluster is assembled. The stepwise computational complexity is simpler in agglomerative clustering than the divisive clustering, but the latter is more useful when one is more interested in the main structure of the data [41] or when the number of clusters (say, k) presented in the data is known in advance [22] (since the stopping criteria for the algorithm can then be modified so that the splitting procedure is no longer performed when k clusters are produced).

However this advantage of the divisive approach is not often helpful in the analysis of gene expression data, because for the visualization of the resulting reorganized microarray data, a full dendrogram is usually desired by biologists. After the full tree is obtained, the determination of the final clusters is achieved by cutting the tree at a certain level or height, which is equivalent to putting a threshold on the pairwise distance between clusters. Note that the final cluster partition is thus rather arbitrary.

Because the agglomerative clustering is more commonly used in microarray data analysis, we will discuss the method in more details. For divisive clustering, see [32, 41] for more information, and for its application on gene expression data, see [4].

Whether agglomerative or divisive, hierarchical clustering in general has several drawbacks. Hierarchical clustering can never repair a decision (to merge in agglomerative clustering, and to split in the divisive one) made in previous steps [41]. It is, after all, based on a stepwise optimization procedure rather than finding k optimal clusters globally. Another disadvantage of hierarchical clustering is that the nature of the hierarchical clustering (where data points are forced into a strict hierarchy of nested subsets) fits more into the context of building a phylogenetic tree [73] rather than that of grouping expression profiles.

Distance measure between two clusters

As we mentioned, in every step of agglomerative clustering, the two clusters that are closest to each other will be merged. Here comes the problem of how we define the distance between two clusters. There are four common options:

1. *Single linkage:* The distance between two clusters is the distance between the two closest data points in these clusters (each point taken from a different cluster).

2. *Complete linkage:* The distance between two clusters is the distance between the two furthest data points in these clusters.

3. *Average linkage:* Both single linkage and complete linkage are sensitive to outliers [22]. Average linkage provides an improvement by defining the distance between two clusters as the average of the distances between all pairs of points in the two clusters.

4. *Ward's method:* At each step of agglomerative clustering, instead of merging the two clusters that minimize the pairwise distance between clusters, Ward's method [84] merges the two clusters that minimizes the "information loss" for the step. The "information loss" is measured by the change in the sum-of-squared-error of the clusters before and after the merge. In this way, Ward's method assesses the quality of the merged cluster at each step of the agglomerative procedure.

These methods yield similar results if the data consist of compact and well-separated clusters. However, if some of the clusters are close to each other or if the data have a dispersed nature, the results can be quite different [22]. Ward's method, although less well-known, often produces the most satisfactory results.

Visualization of the results

A heat map presenting the gene expression data, with a dendrogram to its side indicating the relationship between genes (or experimental conditions) is the standard way to visualize the result of hierarchical cluster analysis on microrray data. The length of a branch in the dendrogram is proportional to the pairwise distance between the clusters. Importantly, the leaves of the dendrogram, and accordingly the rows of the heat map, can be swapped (without actually changing the information contained in the tree) so that the similarity between adjacent genes are maximized, and hence the patterns embedded in the data become obvious in the heat map. However, the time complexity of such an optimal organization of the dendrogram is $O(2^{N-1})$ (because for each of the $N-1$ merging steps there are two possible orders to arrange the concerned clusters). Yet, the structure of the dendrogram remains an important problem, because although the dendrogram itself does not determine the clusters for the users, a good ordering of the leaves can help the users to identify and interpret the clusters. A heuristic approach aiming to find a good solution was developed

[25] by weighting genes using combined source of information, and then placing the genes with lower average weight earlier in the final ordering. Further, [10] reported a dynamic programming method that helps to reduce the time and memory complexities for solving the optimal leaf-ordering problem.

3.2.2 K-means clustering

K-means clustering [32] is a simple and widely used partitioning method for data analysis. Its helpfulness in discovering groups of coexpressed genes has been demonstrated [75].

The number of clusters k in the data is needed as an input for the algorithm. The algorithm then initializes the mean vector for each of the k clusters either by hard assignment (e.g., from the input, or by random generation). These initial mean vectors are called the seeds. Next, the k-means algorithm proceeds iteratively with the following two steps (1) using the given mean vectors, the algorithm assigns each gene (or experiment) to the cluster represented by the closest mean vector, (2) the algorithm recalculates the mean vectors (which are the sample means) for all the clusters. The iterative procedure converges when all the mean vectors of the clusters remain stationary.

From the above description, k-means method can be understood as a distance-based approach. Both the Pearson correlation and the Euclidean distance can be used as the distance measure. However, the squared Pearson correlation metric should not be used in combination with the k-means algorithm, since this distance measure takes the inverse relationships between the clustering subjects into account, which can lead to problems when calculating the means. On the other hand, k-means algorithm is closely related to model-based methods. When the Euclidean distance metric is used, the step of recalculating the means of the clusters actually corresponds to minimizing the within-cluster sum of squared distances from the cluster mean [79]. We will show in Section 3.3.1 that in this case the k-means algorithm is an approximation to the expectation-maximization (EM) method for Gaussian mixture model parameterization [8].

A significant problem associated with k-means algorithm is the arbitrariness of predefining the number of clusters, since it is difficult to predict the number of clusters in advance. In practice, this implies the use of a trial-and-error approach where a comparison and biological validations of several runs of the algorithm with different parameter settings are necessary [55]. Another parameter that will influence the result of k-means clustering is the choice of the seeds. The algorithm suffers from the problem of converging to local minima. This means that with different seeds, the algorithm can yield very different result. Preferably, the seeds should be chosen close to the center of the natural clusters. Of course, this is hard to achieve if no prior knowledge about the clusters is available, which is often the case. Using principle component analysis (PCA) to provide prior knowledge on the number and the means of the clusters was proposed in [58].

3.2.3 Self-organizing maps

SOM [45] is a technique to visualize the high-dimensional input data (in our case, the gene expression data) on an output map of neurons, which are sometimes also called nodes. The map is often presented in a two-dimensional grid (usually of hexagonal or rectangular geometry) of neurons. In the high-dimensional input space, the structure of the data is represented by prototype vectors (serving similar functions as the mean vectors in the k-means algorithm), each of which is related to a neuron in the output space.

As an input for the algorithm, the dimension of the output map (e.g., a map of 6×5 neurons) needs to be specified. After initializing the prototype vectors, the algorithm iteratively performs the following steps. (1) Every input vector (e.g., representing a gene expression profile) is associated with the closest prototype vector, and thus is also associated with the corresponding neuron on the output space. (2) Update the coordinates of a prototype vector based on a weighted sum of all the input vectors that are assigned to it. The weight is given by the neighborhood function (a kernel function in nature), which can be a Gaussian distribution function, applied in the output space. That is, in the updating step, a prototype vector is pulled more toward input vectors that are closer to the prototype vector itself and is less influenced by the input vectors located farther away. In the meantime, this adaption procedure of the prototype vectors is reflected on the output nodes – nodes associated with similar prototype vectors are pulled closer together on the output map. (3) To put a simulated annealing kind of flavor, the initial variance of the Gaussian neighborhood function is chosen so that the neighborhood covers all the neurons, but then the variance is decreased during every iteration so as to achieve a smoother mapping. The algorithm terminates when convergence of the prototype vectors is achieved.

Instead of the batch procedure described above, there is also an online procedure for SOM training. The only difference is, in the online procedure, a random input vector is picked one at a time at the start of the iteration and all the prototype vectors (together with the neurons) are then adjusted accordingly.

From the cluster analysis point of view, SOM methods looks similar to k-means methods. SOM clustering differs from k-means clustering in that a cluster has two "faces" in an SOM – it is represented by the prototype vector in the input space and the neuron on the output space. In this way, an SOM provides a direct means to visualize relations among different clusters. Moreover, a prototype vector is adjusted according to not only the data points that are associated with it but also data points that are assigned to other prototype vectors. SOM clustering is reported to have satisfactory results on gene expression data [73, 78]. In these experiments, every neuron represents a cluster. As a result, clusters that represents similar gene expression profiles are located closer on the output map, while clusters with anti-correlated expression profiles are put into opposite corners of the grid [68].

Because of the advantage in visualization, choosing the geometry of nodes for an SOM is not as crucial a problem as the choice of the number of clusters for a

k-means method. Of course, initializing an SOM with too few nodes will result in non-representative and non-distinctive clusters. However, on the contrary, if too many nodes are added to an SOM, clusters with great similarity (located in the same neighborhood on the output map) can be merged to get a more extensive cluster. Based on this idea, a tree-structured SOM clustering method is implemented for gene expression data in [78]. Like the k-means method, the initial choice of prototype vectors remains a problem that influence the final clustering result of SOM clustering. A good way to seed the prototype vectors can be the result from a PCA analysis [45].

3.3 Model-based approaches for gene expression data

Model-based clustering [32] is an approach that is not really new and has already been used in the past for other applications outside bioinformatics. However, its potential use for cluster analysis of gene expression profiles has been proposed only recently [93, 31, 52],comparing with those are introduced in Section 3.2.

Model-based clustering assumes that the data are generated by a finite mixture of underlying probability distributions, where each distribution represents one cluster. The problem, then, is to associate every gene (or experiment) with the best underlying distribution in the mixture, and at the mean time, to find out the parameters for each of these distributions.

Regardless of the choice of underlying distributions, a mixture model is usually learned by an expectation-maximization (EM) algorithm. Given the microarray data and the current set of model parameters, the probability to associate a gene (or experiment) to every cluster is evaluated in the E step. Then, the M step finds the parameter setting that maximizes the likelihood of the complete data. The complete data refers to both the (observed) microarray data and the assignment of the genes (or experiments) to the clusters. The likelihood of the model increases as the two steps iterates, and convergence is guaranteed.

In the following, we discuss respectively the work of Yeung *et al.* [93] where a mixture model of normal distributions is used for the cluster analysis of microarray data, and the work of McLachlan *et al.* [52] who use a mixture of factor analysis for clustering experiments.

3.3.1 Mixture model of normal distributions

When multivariate normal distributions are used, each cluster is represented by a hypersphere or a hyperellipse in the data space. The mean of the normal distribution gives the center of the hyperellipse, and the covariance of the distribution specifies its orientation, shape, and volume. The covariance matrix for each cluster can be represented by its eigenvalue decomposition, with the eigenvectors determining the orientation of the cluster, and the eigenvalues specifying

the shape and the volume of the cluster. By using different levels of restrictions on the form of the covariance matrix (i.e., its eigenvectors and eigenvalues), one can control the trade-off between model complexity (the number of parameters to be estimated) and flexibility (the extent to which the model fits the data).

The choice of the normal distribution is partly based on its desirable analytic convenience. Moreover, the assumption for fitting normal distribution to gene expression profiles is considered to be reasonable especially when the proper pre-processing procedures (see Chapter 2) have been applied [93, 7]. Of course, other underlying distributions, such as gamma distributions or mixtures of Gaussian and gamma distributions, can also be used to describe expression profiles [87]. So far, no precise conclusions have been made on what is the most suitable distribution for gene expression data [7].

The EM procedure is repeated for different numbers of clusters and different covariance structures. The result of the first step is thus a collection of different models fitted to the data and all having a specific number of clusters and specific covariance structure. Then, the best model with the most appropriate number of clusters and covariance structure in this group of models is selected. This model selection step involves the calculation of the Bayesian information criterion (BIC) [64] for each model.

Yeung *et al.* [93] reported good results of such analysis as described above using their MCLUST software [26] on several synthetic data sets and real expression data sets.

Relation of normal mixture learning with k-means approach

The k-means algorithm described in Section 3.2.2 can be viewed as a special case of applying EM algorithm to learn a mixture model of normal distributions where all the normal distributions in the mixture are characterized by spheres of the same volume but different means (the covariance matrices have the same form of an identity matrix multiplied by a constant). The E step of an EM algorithm is to evaluate the conditional probability of the latent variable of a data point (the latent variable indicates to which cluster the data point belongs) based on the observed data (in our case, the microarray data) and the current parameters. In this context, k-means method uses an index function (i.e., a distribution function of zero variance) to replace the conditional distribution and thus performs a hard assignment to put a data point in the cluster whose mean (i.e., center) is the closest. The next iterative step of k-means clustering is to minimize the cost function of sum-of-squared-distance within the cluster. Using this index function as the conditional distribution, the minimum of this cost function is exactly the maximum of the log-likelihood function [32]. Thus, this second step of k-means corresponds to the M step in the EM algorithm in this case. The parameters from the point of view of the optimization are the sample means of the data points of the clusters.

3.3.2 Mixture model of t distributions and mixture of factor analysis

For the clustering experiments (e.g., tissue samples), however, problem rises for fitting a normal mixture to the data because the number of genes is much larger than the number of experiements. To solve this problem, [52] applied mixture of factor analysis to the clustering of experiments (see Figure 3.1). The idea can be interpreted as follows. A single factor analysis performs a dimensional reduction in the gene space of a cluster. That is to say, in factor analysis, vectors of experiments located in the original n-dimensional hyperellipse (where n represents the number of genes) are projected onto their corresponding vectors of factors located in an m-dimensional unit sphere (usually $m << n$). By using a mixture of factor analysis, clustering of the experiments is done on a reduced feature space (i.e., the m-dimensional factor space) instead of on the original huge dimensional gene space. The EM algorithm is also used to learn the mixture of factor analysis model.

However, the choice for the number of factors in such a model remains a dilemma. If the number is too small, the full correlation structure of the genes cannot be captured; while if it is too large, the EM algorithm for the parameterization of the model can encounter computational difficulties. To alleviate the problem, [52] added another stage to reduce the dimension of the gene space before applying the mixture of factor analysis to the clustering of the experiements. In this stage, both a two-component mixture model of univariate t distributions (where the association of the experiments to the two components is unknown) and a single t distribution are fit to the data for each gene. A threshold on the likelihood ratio between the two models is then applied to determine whether the gene is responsible for the clustering of experiments.

A t mixture model is more suitable for describing a gene expression profile than a normal mixture model because the former is more robust to outliers. A t distribution has an additional parameter called the degree of freedom compared to a normal distribution. The degree of freedom can be seen as a parameter for adjusting the thickness of the tail of the distribution. A t distribution with a relative small degree of freedom will have a thicker tail than a normal distribution with the same mean and variance. However, as the degree of freedom goes to infinity, the t distribution approaches the normal distribution. Because of the thicker tail of a t distribution, the model learned for the t mixture is more robust to the outliers in gene profiles. Therefore, the degree of freedom can be viewed as a robustness tuning parameter.

Their software EMMIX-GENE (where these methods are implemented) yields promising results on several biological data sets.

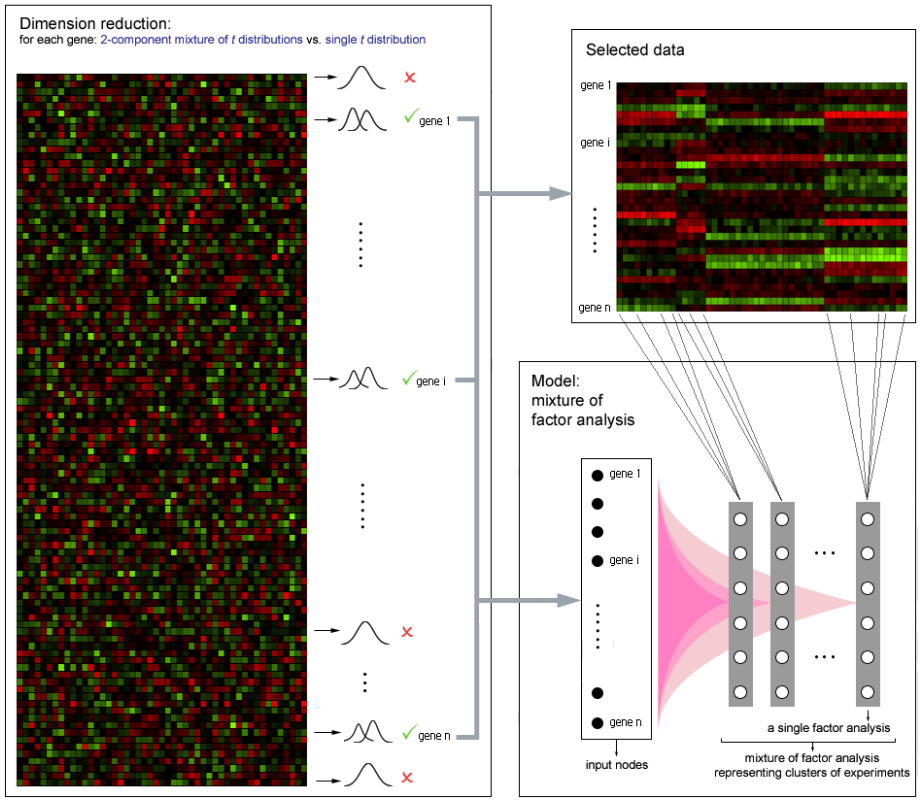


Figure 3.1: McLachlan *et al.* (2002) uses a two-component mixture model of t distributions to examine every gene expression profile against a single t distribution. Expression profiles to which the mixture models fit better (in terms of, for example, likelihood) are selected for further analysis. A mixture of factor analysis is applied on the selected data to cluster the experimental conditions.

3.4 Biclustering algorithms

In addition to discovering the relationship between the genes or the relationships between the the experiments as in the conventional clustering algorithms, biclustering methods also explore the relationships between the genes and the experiments. The prefix in the word biclustering indicates that this is a technique to cluster both the genes and the experiments at the same time.

Although the existing biclustering algorithms originate from the same idea, they differ from each other by their particular emphasis on the problems they try to solve. Among early papers on biclustering methods, clustering algorithms are applied (iteratively) to both dimension of a microarray data set. As the result, genes and experiments are reorganized so as to improve the manifestation of the patterns inherited in both the genes and the experiments. In other words, these algorithms divide the data into checkerboard units of patterns. Examples of these algorithms are the works by Alon *et al.* [4] and Getz *et al.* [30], which exploit existing clustering algorithms for the task. In addition, there are also algorithms particularly designed for this purpose. In [47], mixtures of normal distributions, which is called the plaid model by the authors, are used to describe the microarray data and EM is applied for parameter estimation. For another example, the spectral biclustering method [44] applies singular value decomposition for solving the problem.

However, this type of biclustering algorithm has its limitation when the expression profiles of some genes under study divides the samples in correspondence with one biological explanation (say, tumor type) while profiles of another subset of the genes divides the samples according to another biological process (e.g., drug response) [33]. The second type of biclustering algorithm aims to find genes that are responsible for the classification of the samples. Examples are the gene shaving method [33], which searches for clusters of genes that vary as much as possible across the samples with the help of PCA, and a minimum description length method [40] that identifies gene clusters responsible for classification of experimental conditions.

The third type of biclusteirng algorithm questions conventional clustering algorithms by the idea that genes that share functional similarities do not have to be coexpressed over all the experimental conditions under study. Instead of clustering genes based on their overall expressional behavior, these algorithms look for patterns where genes share similar expressional behavior over only a subset of experimental conditions. The same idea can be used for clustering the experimental conditions. Suppose a microarray study is carried out on tumor samples of different histopathological diagnosis. The problem then is to find tumor samples that have similar gene expression level for a subset of genes (so as to obtain an expressional fingerprint for the tumor). To distinguish the two orientations for this type of biclustering problem, we will refer to the former case as biclustering genes, and the latter case as biclustering experiments. This type of biclustering algorithms is pioneered by Cheng and Church (2000) [19],

where a heuristic approach is proposed to find patterns as large as possible that have minimum the mean squared residues while allowing variance to be present across the experiments when biclustering genes (or across the genes when biclustering experiments). This dissertation is dedicated to the tackling of the problem by using model based approaches, where the parameterization is done by Gibbs sampling. Another model-based approache is provided by Barash and Friedman (2002) [11], where EM algorithm is used for estimating the parameters of the model.

The work of Segal *et al.* [66] brings the third type of biclustering problem to a higher level. In their paper, they are interested in identifying not only the relevant experimental conditions for which the relation between genes of a potential group exists but also the significant attributes associated with the genes and the conditions that are responsible for the generation of such patterns. The method incorporates additional information of the attributes (for example, for a gene, the attributes could be functional role, cellular location or the transcriptional factor (TF) binding sites in gene's promoter region, and for a experimental condition, they could be tumor type, or gene knock out information) together with the gene expression data in a probabilistic framework – the probabilistic relational models (PRMs) [28] in particular, which is an extension of Bayesian networks – and uses (structral) EM [27] for the inference for the parameters of the probabilistic models. In [65], the capability of this framework is illustrated in unveiling the regulation programs of the genes from gene expression data.

Madeira and Oliveira (2004) [51] gives a comprehensive survey on the existing biclustering algorithms. In the following of the chapter, we discuss the gene shaving algorithm [33], the biclustering algorithm of Cheng and Church (2000) [19], and the PRMs for microarray data [66, 65] in more detail.

3.4.1 Gene shaving

As we mentioned previously, gene shaving is an algorithm that tries to find a small subset of genes that exhibit the largest variations across the experimental conditions. The intuition is that these genes may help in explaining the classification of the experiments. To search for a gene cluster, the algorithm performs a PCA on the p-dimensional space, where p stands for the number of experiments in the data set. The largest principle component, called the eigengene, points to the direction where the cloud of data points (representing the genes) expands with largest variation. Then the correlation between every gene with this eigengene is calculated. Genes with the smallest (absolute value of) correlation are "shaved off" (discarded). The proportion of the genes to be shaved off is typically 10%. This procedure is carried on until there is only one gene left in the data set. The remaining data set at every step is treated as a candidate gene cluster. To determine the output cluster from these candidate clusters, the ratio of the within-variance versus the between-variance of every candidate cluster is calculated. The within-variance of a cluster measures the variability of each gene about the cluster average, and the between-variance of a cluster

examines the variance of a cluster across the experiments. The ratio of a candidate cluster is compared with the average ratio of the clusters (which contain the same number of genes) obtained from randomized data sets. The candidate cluster with the largest difference in the ratio is selected as the output cluster. Then, the expression profiles in the whole data set is orthogonalized with the mean of the expression profiles in the output cluster so as to encourage the discovery of an uncorrelated second cluster. The procedure restarts searching for the next cluster.

Hastie *et al.* [33] also shows that the algorithm can also be extended to a supervised form for the discovery of genes related with particular sample classification. Results in the paper illustrate a successful application of the algorithm on predicting patient survival.

3.4.2 Cheng and Church's approach

The biclustering algorithm of Cheng and Church (2000) [19] is a greedy search algorithm to find a set of genes that behave consistently under a subset of conditions.

The consistency of the bicluster is measured by the score of mean squared residue $H(I, J)$. The residue of element a_{ij} in the bicluster indicated by the subsets of I and J is

$$a_{ij} - aiJ - a_{Ij} + aIJ \tag{3.1}$$

where aiJ is the mean of the i^{th} row in the bicluster, a_{Ij} is the mean of j^{th} column in the bicluster, and a_{IJ} is that of all the elements in the bicluster. In order to encourage the gene expression profiles in a bicluster to have fluctuation across the experimental conditions, (because expression profiles with little fluctuation—called trivial biclusters—are not interesting for biologists), they devised another score of row variance as an accompanying score to reject trivial biclusters

$$V(I, J) = \frac{1}{|J|} \sum_{j \in J} (a_{ij} - aIj)^2. \tag{3.2}$$

The goal of the algorithm is thus to find a bicluster with the largest area (i.e. $|I| \cdot |J|$) whose mean squared residue score satisfies $H(I, J) <= \delta$, where δ is a threshold input by the user. The bicluster is rejected if the row variance score $V(I, J)$ is lower than another threshold. Because the optimization problem is an NP hard one, they opted for a greedy search algorithm that starts from the original microarray data set, and delete a few rows or a few columns at a time in the direction to decrease $H(I, J)$. Once the condition $H(I, J) <= \delta$ is reached, there is a chance that the resulting bicluster is not maximal. Thus, another step to add some rows or columns without increasing the score is performed.

The algorithm above aims to find one bicluster at a time for a microarray data set. In order to discover multiple biclusters, the authors mask the found bicluster by replace the matrix entries in the bicluster by random values. However, the masks are not used for the rows/colums addition procedure.

3.4.3 Probabilistic relational models for microarray data

In the preparation to address the problem in the language of PRMs [28], Segal *et al.* [66] first define the relational schema as follows. The objects – genes, experiments, and gene expressions – are associated with their corresponding attributes. Among other attributes such as the TF binding sites for a gene and the knock-out information for an experiment, the class of a gene is one of the attributes of the gene; similarly, the class of an experiment is an attribute for the experiment; and for the expression of a specific gene in a particular array, the expression level is one of its attributes. The values of some of these attributes are given by the data, for instance, the TF binding sites, the knock-out information, and the expression level. However, the values of the other attributes are unknown. The particular examples of interest in this case are the class attributes of the genes and the experiments. The task now is not only to infer the unknown value of the attributes (such as the class attributes so as to obtain the biclusters) but also to find the relationships between all the attributes presented in the problem.

A PRM itself can be viewed as the bigger frame defining the relationships between classes of different objects. In our case, the frame is that gene expressions are influenced by both the genes and the experiments. However, when it comes to a particular case, where the given data includes the microarray data and accompanied information of the genes and the experiments, the PRM has to generate a more detailed probability model, namely a Bayesian network, for the specific problem. First, to put them in a probabilistic language, all the attributes are described as random variables. A Bayesian network is set up to address the dependency structures between the random variables (i.e., the attributes). A Bayesian network [34] is a graphical model where nodes (each representing a random variable) are connected with each other by directed edges. The direction of an edge between two nodes (pointing from a parent node to a child node) indicates the influence of one random variable (the parent node) to the other (the child node). The influence is quantified by the probabilistic distribution of the child node conditioned on the value of the parent node – the conditional probability distribution (CPD). A node is conditionally independent of all the other nodes given the value of its parents. For the microarray data, one can suppose that the attribute of expression level is always placed at the bottom of the Bayesian network structure; i.e., it is always a child node but never a parent node.

To learn the relationships between different attributes now means to decide the structure of the Bayesian network (i.e., to draw the edges) and the CPDs between the nodes. In addition, the learning task also includes the estimation of the unknown values (or called missing values) of some of the attributes. The learning procedure is carried out by the Structural EM algorithm [27]. Briefly speaking, it iteratively executes a structural learning step and an EM step till convergence. In the structural learning step, a search algorithm is performed based on the current estimate of network parameters (CPDs) and missing values

to find the structure that maximizes a Bayesian information score [64]. The EM step uses the learned structure to estimate its CPDs (the M step) and the missing values (the E step).

The efficiency of the method is illustrated in [66] on two yeast data sets as well as synthetic data sets. In [65], it is shown that the method can be tailored for unveiling the regulatory program of the genes.

3.5 Assessing cluster quality

Clustering will produce different results. Even random data often produce clusters depending on the specific choice of preprocessing, algorithm, and distance measure. Therefore, validation of the relevance of the cluster results is of utmost importance. Validation can be either statistical or biological. Statistical cluster validation can be done by assessing cluster coherence, by examining the predictive power of the clusters, or by testing the robustness of a cluster result against the addition of noise.

Alternatively, the relevance of a cluster result can be assessed by a biological validation. Of course it is hard, not to say impossible, to select the best cluster output, since "the biologically best" solution will be known only if the biological system studied is completely characterized. Although some biological systems have been described extensively, no such completely characterized benchmark system is now available. A common method to biologically validate cluster outputs is to search for enrichment of functional categories within a cluster. Detection of regulatory motifs (see [75]) is also an appropriate biological validation of the cluster results. Some of the recent methodologies described in literature to validate cluster results will be highlighted in the following.

1. *Testing cluster coherence:* Based on biological intuition, a cluster result can be considered reliable if the within-cluster distance is small (i.e., all genes retained are tightly coexpressed) and the cluster has an average profile well delineated from the remainder of the data set (maximal intercluster distance). Such criteria can be formalized in several ways, such as the sum-of-squares criterion of k-means [79], silhouette coefficients [41], or Dunn's validity index [6]. These can be used as stand alone statistics to mutually compare cluster results. They can also be used as an inherent part of cluster algorithms, if their value is optimized during the clustering process.

2. *Figure of Merit:* FOM [94] is a simple quantitative data-driven methodology that allows comparisons between outputs of different clustering algorithms. The methodology is related to the jackknife and leave-one-out cross-validation. The method goes as follows. The clustering algorithm (for the genes) is applied to all experimental conditions (the data variables) except for one left-out condition. If the algorithm performs well,

we expect that if we look at the genes from a given cluster, their values for the left-out condition will be highly coherent. Therefore, we compute the FOM for a clustering result by summing, for the left-out condition, the squares of the deviations of each gene relative to the mean of the genes in its cluster for this condition. The FOM measures the within-cluster similarity of the expression values of the removed experiment and therefore reflects the predictive power of the clustering. It is expected that removing one experiment from the data should not interfere with the cluster output if the output is robust. For cluster validation, each condition is subsequently used as a validation condition, and the aggregate FOM over all conditions is used to compare cluster algorithms.

3. *Sensitivity analysis:* Gene expression levels are the superposition of real biological signals and experimental errors. A way to assign confidence to a cluster membership of a gene consists in creating new in silico replicas of the microarray data by adding to the original data a small amount of artificial noise (similar to the experimental noise in the data) and clustering the data of those replicas. If the biological signal is stronger than the experimental noise in the measurements of a particular gene, adding small artificial variations (in the range of the experimental noise) to the expression profile of this gene will not drastically influence its overall profile and therefore will not affect its cluster membership. In this case, the cluster membership of that particular gene is robust with respect to sensitivity analysis, and a reliable confidence can be assigned to the clustering result of that gene. However, for genes with low signal-to-noise ratios, the outcome of the clustering result will be more sensitive to adding artificial noise. Through some robustness statistic [14], sensitivity analysis lets us detect which clusters are robust within the range of experimental noise and therefore trustworthy for further analysis.

   The main issue in this method is to choose the noise level for sensitivity analysis. Bittner et al. [14] perturb the data by adding random Gaussian noise with zero mean and a standard deviation that is estimated as the median standard deviation for the log-ratios for all genes across the experiments. This implicitly assumes that ratios are unbiased estimators of relative expression, yet reality shows often otherwise.

   The bootstrap analysis methods described by Kerr and Churchill [42] to identify statistically significant expressed genes or to assess the reliability of a clustering result offers a more statistically founded basis for sensitivity analysis and overcomes some of the problems of the method described by Bittner et al. [14]. Bootstrap analysis uses the residual values of a linear analysis of variance (ANOVA) model as an estimate of the measurement error. By using an ANOVA model, nonconsistent measurement errors can be separated from variations caused by alterations in relative expression or by consistent variations in the data set. These errors are assumed to be independent with mean zero and constant variance σ^2 but no explicit assumption on their distribution is made. The residuals are subsequently

used to generate new replicates of the data set by bootstrapping (adding residual noise to estimated values).

4. *Use of different algorithms:* Just as clustering results are sensitive to adding noise, they are sensitive to the choice of clustering algorithm and to the specific parameter settings of a particular algorithm. Many clustering algorithms are available, each of them with different underlying statistics and inherent assumptions about the data. The best way to infer biological knowledge from a clustering experiment is to use different algorithms with different parameter settings. Clusters detected by most algorithms will reflect the pronounced signals in the data set. Again statistics similar to that of Bittner et al. [14] are used to perform these comparisons.

   Biologists tend to prefer algorithms with a deterministic output, since this gives the illusion that what they find is "right". However, nondeterministic algorithms offer an advantage for cluster validation, since their use implicitly includes a form of sensitivity analysis.

5. *Enrichment of functional categories:* One way to biologically validate results from clustering algorithms is to compare the gene clusters with existing functional classification schemes. In such schemes, genes are allocated to one or more functional categories [31, 75] representing their biochemical properties, biological roles, and so on. Finding clusters that have been significantly enriched for genes with similar function is proof that a specific clustering technique produces biologically relevant results.

   Using the cumulative hypergeometric probability distribution, we can measure the degree of enrichment by calculating the probability or P-value of finding by chance at least k genes in this specific cluster of n genes from this specific functional category that contains f genes out of the whole g annotated genes

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}} = \sum_{i=k}^{\min(n,f)} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}}.$$

   These P-values can be calculated for each functional category in each cluster. Note that these P-values must be corrected for multiple testing according to the number of functional categories.

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}} = \sum_{i=k}^{\min(n,f)} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}}.$$